Amendments to the Specifications:

Please amend the paragraph beginning at page 1, line 6 as follows:

Secure chips which follow the Trusted Computing Platform Alliance (TCPA) protocols are well known in the art. In ~~this~~ the TCPA specification, a "secure chip" is a Trusted Platform Module (TPM). Typically, the TPM resides in a client computer system in a computer network. Among other functions, the TPM generates encryption keys in the form of public/private key pairs for the client to be used on the network. When the keys are not in use, they are stored outside of the TPM in a secure manner in a "daisy chain" fashion.

Please amend the paragraph beginning at page 2, line 7 as follows:

Keys can be of two types according to the TCPA specification: migratable and non-migratable. Migratable keys are particularly relevant to the present invention, and thus only they will be described here. The TCPA specification contains two commands for migrating keys from one TPM to another. The first command is a simple re-wrap command, where a user's key is loaded into a TPM, unwrapped with its parent's key and then re-wrapped with another parent's key. This command can be used for migrating the user's key from one computer system to another during a computer upgrade. The second command is used for storing the user's key with a third party in case of hardware failure. For the second command, it is not known what the parent key of the replacement system will be during the storage, so a third party's public key is used for wrapping.

Please amend the paragraph beginning at page 6, line 3 as follows:

Figure 3 is a flowchart illustrating a preferred embodiment of a method for improved security with a secure chip in accordance with the present invention. First, the secure chip 102 generates a first random number, via step 302. The first random number is used to create a migratable ~~blob~~ keyblob 202, via step 304. The migratable keyblob 202 contains a key, such as the parent key 108. The secure chip 102 then wraps the migratable keyblob 202 with the public key of the key's parent key, via step 306, which is the public key of the grandparent key 106. The secure chip 102 receives a pass phrase for the user of the key 108, via step 308. The secure chip 102 then generates a ~~second~~ pseudo random number based on the pass phrase, via step 310. ~~A third random number is generated based on the second random number, via step 312.~~ Next, a ~~fourth~~ third random number is generated based on the first random number and the ~~third~~ pseudo random number, via step 314. This ~~fourth~~ third random number is stored, via step 316. The migratable keyblob 202 is then migrated from the computer on which the secure chip 102 resides to itself, via step 318. In the preferred embodiment, the method is performed by a software residing on a disk in the computer on which the secure chip 102 also resides.

Please amend the paragraph beginning at page 7, line 3 as follows:

Figure 4 is a flowchart illustrating in more detail the preferred embodiment of the method for improved security with a secure chip in accordance with the present invention. Assume that the secure chip 102 is a Trusted Platform Module (TPM) using the Trusted Computing Platform Alliance (TCPA) protocol. First, a key, such as the parent key 108, is scrambled, via step 402. Next, the random number generator of the TPM 102 generates a first random number, via step 404. The first random number is then XOR'ed with the scrambled parent key 108 to create the migratable keyblob 202, via step 406. The TPM 102 wraps the migratable keyblob 202 with the public key of the parent key's parent key, i.e., the public key of the grandparent key 106, via step 408. Also, a pass phrase for a user of the parent key 108 is received, via step 410. A ~~second~~ pseudo random number is generated by hashing the user's pass phrase and ~~, via step 412. A third random is generated number by~~ applying a mask generation function (MGF) to ~~the second random number and converting it into~~ produce a string ~~with~~ having the same length as the first random number, via step 414. MGF's are well known in the art. The first random number and the ~~third~~ pseudo random number are XOR'ed to generate a ~~fourth~~ third random number, via step 416. This ~~fourth~~ third random number is stored, via step 418. The migratable keyblob 202 is migrated from the computer with the TPM to itself, via step 420.

Please amend the paragraph beginning at page 7, line 20 as follows:

To use the parent key 108, the user enters his/her pass phrase. Figure 5 is a flowchart illustrating how a key secured with the method in accordance with the present invention is obtained. First, the user's pass phrase is received, via step 502. The ~~third~~ pseudo random number is then obtained from hashing the pass phrase and applying the MGF, via step 504. The first random number is then obtained by XOR'ing the ~~third~~ pseudo random number with the stored ~~fourth~~ third random number, via step 506. The first (TPM's) random number and the encrypted migratable keyblob 202 are then sent to the TPM 102, via step 508. The TPM 102 unwraps the encrypted migratable keyblob 202 using its private key, via step 510. The TPM 102 XOR's the migratable keyblob 202 with the first random number to obtain the scrambled parent key 108, via step 512. The TPM 102 can then unscramble the parent key 108, via step 514. Once unscrambled, the key 108 may be used. While with a conventional migratable keyblob, the recovered key 10 is rewrapped into a normal blob and stored in persistent memory, this does not happen with the recovered key 108 in accordance with the present invention. The returned normal blob is discarded instead.